As you look back on this semester, please take a moment to reflect on your experience, recognizing the challenges, successes, and all that you've learned along the way.

Please respond thoroughly to the following questions (likely about 1-2 paragraphs each).

1. What is one program you wrote that you were particularly proud of during this course? Why were you proud? Explain.

I felt the program for the adventure game was my best effort this semester. I feel good that I was able to use the Hash Table or Dictionary Object in my program. I set variables to hold key strings to drive the action and responses for the user. Instead of a deep complex if then else if chain of logic I used the keyed values to lookup cases for responses.

Though my design did work to fit the rubric, it was fairly easy to configure to allow much more complex paths, such as being able to go back and forth between locations or in circles if the user chose options that would result in such a case.

1. How might you use the skills you learned in this course in the future? This might include skills such as programming, problem-solving, debugging, etc.

Though I was already experienced in programming before this course, only not familiar with python. I have partially learned python, also self-studied into the OOP features of the language. I have begun to dig into being able to perform a web calls and ODBC queries as well. Also a good lesson for me this semester, which I still have trouble with is keeping it simple and going for the more straightforward solution.

I have already begun showing my team-mates at work what python can do, and we are looking to use it in efforts to help us manage the complex lab environments for our application that we support. Also, to use it to assist the automation of the building of our application servers. A concern that my management has is how few on the team know python, so I am trying to bridge that gap by teaching the team the language.

1. Describe an experience from this class that has given you confidence that you can learn new programming skills in the future.

One of the hardest things for me was to step back and not go off on a tangent in the requirements. This is a problem I have at work as well. I think that it in part is trying to write a perfect program that does it all, each time. I have started to gain more confidence in backing off to get the requirements as listed done. Then let the discussion of more requirements come later. This has helped me at work as well, as we are to split the workload on multiple team-members and we all have to be in common goal. For the future this opens up me to work better with my team-mates.